Next Word Prediction in Bodhi Language Using LSTM based Approach

Ankush Kumar
Department of Computer Science & Engineering,
School of Engineering & Technology,
Sharda University,
Greater Noida
Email: ecoankush2000@gmail.com

Pushpendra Kumar Mishra
Department of Computer Science & Engineering,
School of Engineering & Technology,
Sharda University,
Greater Noida
Email:pushpendramishra39@gmail.com

Tsering Namgail
Department of Computer Science & Engineering,
School of Engineering & Technology,
Sharda University,
Greater Noida
Email: namgailyokhangpa@gmail.com

Sandeep Kumar
Department of Computer Science & Engineering,
School of Engineering & Technology,
Sharda University,
Greater Noida
Email: sandeep.csengg@gmail.com

Abstract—

Bodhi language is one of the rare languages which is still spoken in the Leh neighbourhood, Ladakh and many Tibetan regions. There is not much linguistic research done in this language. Even google translate does not work on this language. There are various types of other linguistic researches and model available on language like English and some other regional languages like Hindi, Bangla, Ukrainian etc. But there are almost negligible research and models available on Bodhi Language.

In this paper, we proposed a Language Modelling Technique using Long Short Term Memory network (LSTM) which is based on Recurrent Neural Network (RNN), using this machine learning technique we have made a model to predict the next word in bodhi language, when the user will input anything, the model will predict the next word according to the previous word(s). This model is already made for the English language but we are making the model or basically programing the model to predict the next word in the Ladakhi language which is also called as Bodhi language. This language is more complex than English language. We have tried to make the model as accurate as possible while predicting the next word in Ladakhi language. To prepare the

model we have collected dataset as a large collection of Bodhi words. In this model, we have trained the model in 500 iterations (Epochs).

we used the TensorFlow, keras, dictionaries, pandas, NumPy packages. For the coding purpose we used the platform called Google Colab which is provided by google for machine learning enthusiasts.

Keywords: NLP, Next word prediction, RNN, LSTM.

I. INTRODUCTION

We are living in the word of machine learning in which we see different machine learning algorithms for different kind of automated works and services. Machines which work on themselves without any human touch and effort. And automatically improve themselves through experiences of previous and current work. Machines can easily recognize the patterns in between the words and things which humans cannot recognize. We very well know that. In traditional programming, the solution is the output, while the inputs are rules and data.

Whether we're merely sending messages or browsing the internet, we usually text on gadgets like our phones and desktops. While doing such things, we can notice that it

Preprint not peer reviewed

This preprint research paper has not been peer reviewed. Electronic copy available at: https://ssrn.com/abstract=4367666

proposes the next word based on what we are typing. This function, called next word prediction, foresees the word that could appear after one of our messages. It saves us time and enables us to write more swiftly and effectively. The next word prediction (NWP) problem is a significant one in the field of natural language processing. Another name for it is language modelling, and text mining is used in this case.

In this essay, we discuss the Bodhi language, an uncommon tongue that is only spoken in the Ladakh area. It is often written in Tibetan Script, and compared to most other Tibetan dialects, its sound is considerably more similar to the classical Tibetan language. It also goes by the name Bhoti language. Many of the prefix, suffix, and head characters that are silent in many other Tibetic languages, particularly Central Tibetan, are spoken by Ladakhis. The Ladakhi alphabets are shown in figure n. 1 below.

Figure 1. Alphabets in Ladakhi Script

Even Google Interpret does not have the ability to translate the Bodhi language, which is still used in Tibet and the surrounding Nepal. We are the first to attempt next word prediction in the Bodhi language, taking a step towards the strategy of preserving this extremely uncommon language. The programmers who are now working on this LSTM model are aware of how well it predicts the following word. A variant of the recurrent neural network (RNN) architecture is long short-term memory (LSTM). However, because there is no word spacing in Bodhi, the LSTM model does not perform well when applied to this language. As a result, we must provide word space. The developers made their decision to employ LSTM because they thought it may help people remember important terms for a longer period of time. The goal of creating this model is to properly predict the next word based on input in the very language of Bodhi. In this study, we employed a variety of input changes to train the system to recognize patterns and produce correct predictions.

II. LITERATURE REVIEW

One of the current hot areas in natural language processing research is next word prediction. On this subject, numerous study papers have been published. These papers serve as the sources for our study paper.

For the Assamese language, Perth Pratima Barman employed an RNN-based technique to predict the following word [1]. Additionally, because Assamese is a native language, there are numerous translations of a same statement into English. It becomes harder because of this. Because it preserves some memory while being trained, he used an RNN model of the LSTM type. He compared the results of two datasets, one for simple Assamese and the other for Assamese phonetics, to determine which would be more effective for e-communication and save more time while producing more accurate results.

Ilya Sutskever employed RNN in his model in a similar way. A RNN variation that makes advantage of multiplicative or gated connections was employed [2]. Word

Preprint not peer reviewed

This preprint research paper has not been peer reviewed. Electronic copy available at: https://ssrn.com/abstract=4367666

prediction, next sentence choice, and sentence topic prediction are three NLP-based tasks that Shalini Ghosh and her team performed using a contextual LSTM model. And while choosing the next sentence, they saw a considerably higher accuracy of 21% compared to 18%.

Aaron Mueller [3] evaluated various word prediction models using cross-linguistic syntactic analysis. He conducted a comparison study on it and utilized CLAMS to test various neural network models. Furthermore, LSTM assessments for monolingual evaluations were generally accurate and reliable.

Jingyun Yang proposed the MCNN-ReMGU model [4], which combines data regularization technology with residual-connected MGU network and multi-window convolution. Their approach eliminated the gradient vanishing issue and was effectively.

Sourabh Ambulgekar et al. [5] had also researched on the field of next word prediction using RNN. They used LSTM based sequential model with 3 layers. They predicted 10 words accordance to 40 previous words. Omor Faruk Rakib et al. [6] had researched prediction of next word in their language Bangla for which they used GRU which is a modified version of RNN.

Shalini Ghosh et al. [7] introduced power of context to LSTM which is a type of RNN technique. Sakib Haque et al. [11] proposed the problem of action word prediction. Basically, They predicted the action word or verb of the code summary. They used these action word to summarize the code. Mohamed Suliman et al. [12], in their paper two models better than one, raised the concerns regarding privacy on Google's GBoard app which widely used all over the word. In which they showed how word recovery and reconstruction of sentences can be used for reconstruction attacks.

A variety of small and big neural networks with four layers were trained by Joel Stremmel et al. [13] using a decentralised technique. The federated Averaging Algorithm was used to train their models. Lookahead Information Gain (LIG), a measure of word comprehension effort suggested by Aurnhammer et al. [14], was not shown to have any further impacts beyond the unexpected.

In order to anticipate the following word in Hindi, Radhika Sharma et al. [15] employed Bi-LSTM. They discovered that the accuracy of LSTM was 59.46% and that of Bi-LSTM was 81.07%. The dataset created by IIT Bombay for the Indian Language Technology Center was utilised. Hozan K. Hamarashid et al. [16] created an N-gram model for Kurdish Sorani and Kurmanji.

They utilized the N-gram model to suggest text accurately. To do so they build the application using R programming and R studio. Tiwalayo N. Eisape et al. [17] in their paper they tried to improve the neural language models. They used Provo Corpus data. They analyzed various models like N-gram using a window size of 5, LSTM, GPT-2, Transformer-XL etc. They found the LSTM model does not perform as well as Transformer-based model and Gated recurrent unit outperformed transformer models with lower perplexity in predicting gaze duration.

Thus, previously various researchers researched in this context and improved the language modeling to predict next words including some who built the prototype models for their regional and national languages which predicted the next words pretty accurately. Yet we did not find any work on Bodhi language. And hence we made a model which predicts the next word in Bodhi.

III.METHODOLOGY

We used a LSTM based approach to build our model in which we build a sequential model with 4 layers in which two LSTM layers and two Dense layers. The input size of the model is 3 and output is 1 while the sequence length is 10. We trained our model in batch size of 64

Preprint not peer reviewed

This preprint research paper has not been peer reviewed. Electronic copy available at: https://ssrn.com/abstract=4367666

and in 500 epochs. Long Short-Term Memory (LSTM) is a category of RNN (Recurrent Neural Network). It is a particular form of RNN that can identify long-term dependencies in a set of data. This is accomplished through the repeating module's four levels, which work together to interact. The LSTM's basic design is shown in the figure n.2 below.

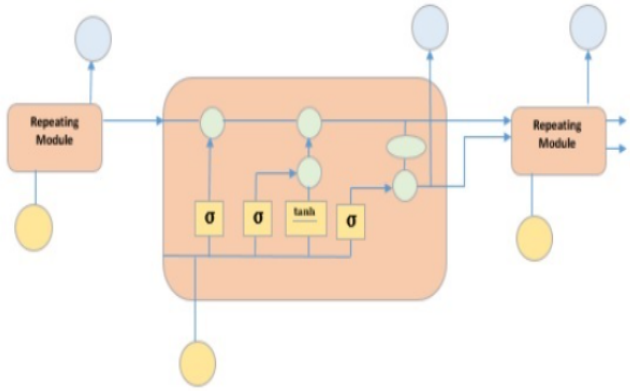

Figure 2. Basic Architecture of LSTM

In the forget gate, the sigmoid function is used. whereas the information flow is controlled by an input gate. The Sigmoid and Tanh functions are multiplied point-wise for the input gate, respectively. While the output gate determines which data should be sent to the following state.

B) Dataset

The dataset was compiled from around 1000 Bodhi words which were from various Ladakhi articles, newspapers, and dictionaries. Sentences made up of words are then combined to serve as the input for training and testing. These sentences have a word count of five or more. Many brief sentences with sequences of less than six words can be found in the dataset. Figure n. 3 shows the snap of the dataset.

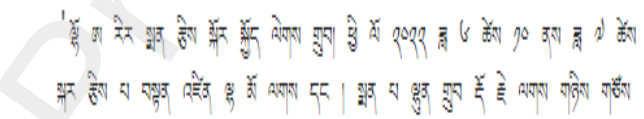

Figure 3. Snap of the dataset

To prepare the model we divided the model into modules which solves the sub problems. The dataflow between these modules of sub-task is given by the diagram which is given in the figure n. 4.

The figure describes how data flows from various sub tasks which performs some operations on data including cleaning and preprocessing of data to training the model and after that the prediction of the next word. It only shows the sub tasks for preparing the model and the dataflow through it. It does not show how many layers are in the model and how and which operations are performed in the subtask.

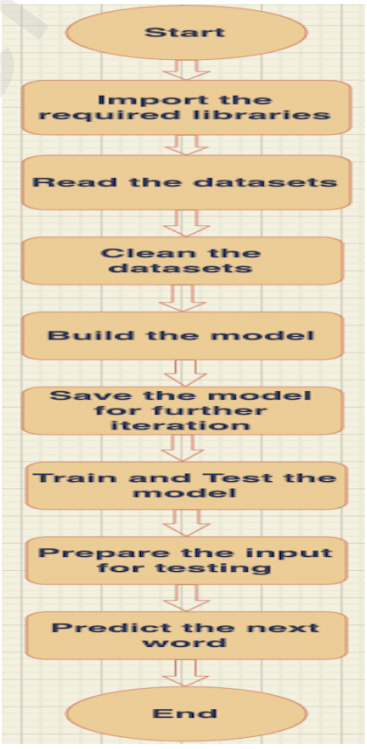

Figure 4. Flowchart of the method

C) Dataset pre-processing

All the processes or operations performed on data before feature engineering comes under the data-preprocessing in which cleaning the dataset, removing outliers and converting the

Preprint not peer reviewed

This preprint research paper has not been peer reviewed. Electronic copy available at: https://ssrn.com/abstract=4367666

data into machine understandable language or in binary digits comes.
Before starting the procedure, the dataset must be cleaned. The words in the dataset are then divided into several groups. The iterator is used to parse the input file and collect distinct words. With the help of tokenizer function in preprocessing library of tensorflow package tokens were generated for sequential text. This demonstrates that words are challenging for machine learning neural networks to process, making it essential to map them to indices, which are straightforward for neural networks to recognize.

The process starts with the sequencing of the given words after the sequencing the feature engineering is done and input and output features were created. For prediction, we used 3 words to predict the next words for which the input was the np array with 4 sequence elements and output was the last element of the np array. We used a Sequential model with 4 layers. Our model was fairly accurate. We used our model to predict in both languages in English as well as bodhi language. Model was fairly accurate in English language but its accuracy decreased in Bodhi language. Yet we tried to raise its accuracy by separating the words and then tokenization of these words.

In English language, we trained our model for only 7 Epochs in last epoch loss was equal to 4.4253 and accuracy was around 70%.

```
Epoch 7/7
1958/1958 [==============================] - ETA: 0s - loss: 4.4253
Epoch 7: loss improved from 4.62284 to 4.42526, saving model to next_words.h5
1958/1958 [==============================] - 30s 15ms/step - loss: 4.4253

Epoch 7/7
1958/1958 [==============================] - ETA: 0s - loss: 4.4253
Epoch 7: loss improved from 4.62284 to 4.42526, saving model to next_words.h5
1958/1958 [==============================] - 30s 15ms/step - loss: 4.4253
```

Figure 5A and 5B. Epochs for next word prediction in English

In Bodhi language we trained our model for 500 iterations and loss was around 0.00040 and accuracy was 50%.

```
Epoch 499/500
105/108 [============================>.] - ETA: 0s - loss: 4.0386e-04 - accuracy: 0.4426
Epoch 499: loss did not improve from 0.00040
108/108 [==============================] - 2s 15ms/step - loss: 4.0322e-04 - accuracy: 0.4432
Epoch 500/500
108/108 [==============================] - ETA: 0s - loss: 4.0279e-04 - accuracy: 0.4439
Epoch 500: loss did not improve from 0.00040
108/108 [==============================] - 2s 15ms/step - loss: 4.0279e-04 - accuracy: 0.4439
```

Figure 6. Epochs for next word prediction in Bodhi

As this language is very hard for preprocessing and training the model, yet we managed to train our model, and test it manually. The loss of our model kept on decreasing in our model. It was very hard to collect dataset for training our model and in future as dataset increases our model will improve in accuracy as well as performance. We prepared a separate function for preparation of user input. This function takes input from user tokenize it and preprocessing happen which converts the input text into tokens with indexing of serial numbers. Then the input is given to model. Model works on input and gives output as a word.

IV. RESULTS AND DISCUSSIONS

While training our model the loss kept on decreasing. Loss to Epochs graph is given below which visualizes how the loss is behaving in each epoch. The figure given below visualizes the trend of loss during training (figure 8 shows).

Preprint not peer reviewed

This preprint research paper has not been peer reviewed. Electronic copy available at: https://ssrn.com/abstract=4367666

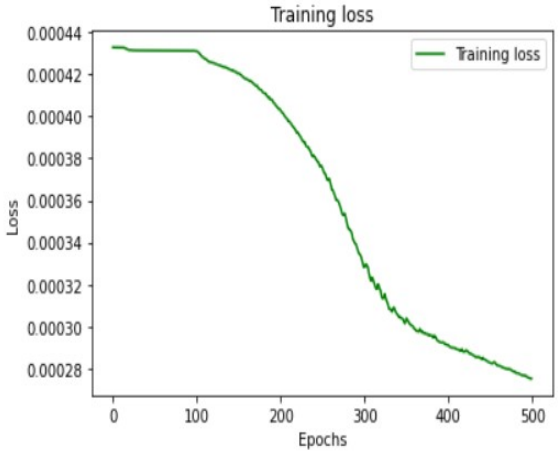

Figure 7. Training loss

Accuracy trends of our model throughout all training epochs which is visualized by the figure n. 9 given below

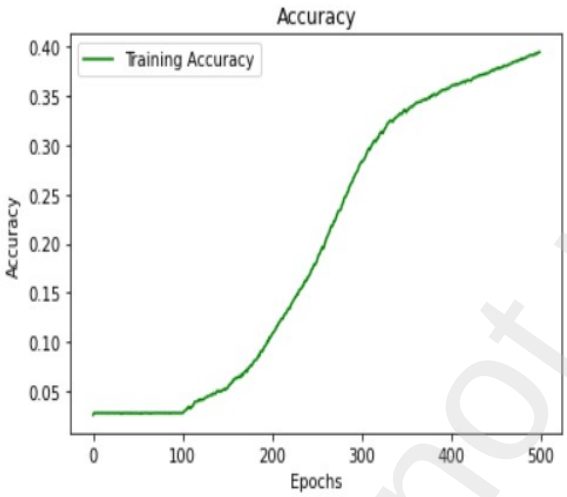

Figure 8. Accuracy.

We created a function for preparing input and feeding it to our model. With the help of the tokenization our input text was sequenced. our model processes the input and gives one and the most probable word as output. Accuracy of our model is 50%. Now our model can predict the next word in Bodhi language. Which is the language of Ladakh and surrounding regions including the parts of Nepal and Tibet. We used 3 words as input and accuracy was fairly high in comparison to 1 word as input which is 50%. A snap of the output of the model is given below in figure n. 9.

```
Enter the line: ཐ་མ་ད་ན།
['ཐ', 'མ', 'དན']
1/1 [==============================] - 1s 745ms/step
next suggested word is:  པའི
Enter the line: རྒྱ་གར་སྐྱོ
['རྒྱ', 'གར', 'སྐྱོ']
1/1 [==============================] - 0s 18ms/step
next suggested word is:  ཤིང
Enter the line: འཇིག་རབ་སྒྲུབ་རྟེན་གང
['སྒྲུབ', 'རྟེན', 'གང']
1/1 [==============================] - 0s 18ms/step
next suggested word is:  གི
Enter the line: བཙན་བྱོལ་ཁ་དུ་འབྱོར་རྒྱལ་མཚན་པོ་གང
['མཚན', 'པོ', 'གང']
1/1 [==============================] - 0s 18ms/step
next suggested word is:  རྫོང
Enter the line: 0
Execution completed....
```

Figure 9. Output with 3 words as input

V. CONCLUSION

Next word prediction is one of the most researched NLP fields because it is about finding text. We have used an LSTM model which is trained in 500 iterations. Our work is the first to be done on a rare language like Bodhi. From the result, it can be said that the accuracy is sufficiently high. This model can be used to predict the next word from the target's input. This model works on a language which even google translate does not provide a service for, which is a feat in itself.

VI. FUTURE SCOPE

As the dataset was pretty hard to collect, hence in the future we can gather more data so that our model can be trained thoroughly. And its accuracy can be increased. There is a need to research the proper method for the preprocessing of dataset in Bodhi language so that accuracy can be increased.

Preprint not peer reviewed

This preprint research paper has not been peer reviewed. Electronic copy available at: https://ssrn.com/abstract=4367666

REFERENCES

[1] Barman, P. P., & Boruah, A. (2018). A RNN based Approach for next word prediction in Assamese Phonetic Transcription. *Procedia computer science*, *143*, 117-123.

[2] Sutskever, I., Martens, J., & Hinton, G. E. (2011, January). Generating text with recurrent neural networks. In *ICML*.

[3] Mueller, A., Nicolai, G., Petrou-Zeniou, P., Talmina, N., & Linzen, T. (2020). Cross-linguistic syntactic evaluation of word prediction models. *arXiv preprint arXiv:2005.00187*.

[4] Yang, J., Wang, H., & Guo, K. (2020). Natural Language Word Prediction Model Based on Multi-Window Convolution and Residual Network. *IEEE Access*, *8*, 188036-188043.

[5] Ambulgekar, S., Malewadikar, S., Garande, R., & Joshi, B. (2021). Next Words Prediction Using Recurrent NeuralNetworks. In *ITM Web of Conferences* (Vol. 40, p. 03034). EDP Sciences.

[6] Rakib, O. F., Akter, S., Khan, M. A., Das, A. K., & Habibullah, K. M. (2019, December). Bangla word prediction and sentence completion using GRU: an extended version of RNN on N-gram language model. In *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)* (pp. 1-6). IEEE.

[7] Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., & Heck, L. (2016). Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.

[8] Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., ... & Fernández, R. (2016). The LAMBADA dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

[9] Ranzato, M. A., Chopra, S., Auli, M., & Zaremba, W. (2015). Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

[10] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

[11] Haque, S., Bansal, A., Wu, L., & McMillan, C. (2021, March). Action word prediction for neural source code summarization. In *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)* (pp. 330-341). IEEE.

[12] Suliman, M., & Leith, D. (2022). Two Models are Better than One: Federated Learning Is Not Private For Google GBoard Next Word Prediction. *arXiv preprint arXiv:2210.16947*.

[13] Stremmel, J., & Singh, A. (2021, April). Pretraining federated text models for next word prediction. In *Future of Information and Communication Conference* (pp. 477-488). Springer, Cham.

Preprint not peer reviewed

This preprint research paper has not been peer reviewed. Electronic copy available at: https://ssrn.com/abstract=4367666